03/04/2020

* Whenever you will create an Angular project by default component it will hold is "App"

* Every Angular project will have one by default module. i.e app.module.
* Every component has to registered with module.

V HelloApp (Projectname)
> e2e (End to end basically for testing purpose)
> node_modules (All the 3rd party apps will be stored here)
V src (In this you can have a number of modules and it can hold multiple number of components)

V app (by default module)
TS app-routing.module.ts
app.component.css
<> app.component.spec.ts
TS app.component.ts
TS app.module.ts

> assets (for storing images)
> environment (for configuration setting)
* favicon.ico (for icon)
<> index.html (responsible for displaying the code on browser)
TS main.ts (it's ts code it's entry point of our program)
Style.css (globally if you want to do style you can go with this)

Instead of going component.css in each you can go with style.css.

Structure of Angular

1) e2e : a) This is where we write end to end test for our application

2) Node - modules: We will store all 3rd party application.

3) Source (src) : In the src folder we have app(module) Inside that all component will be present.

Each angular project will have one component i.e "app component" and one module i.e "app.module.ts

4) assed: This is where we store image files.

5) environments: This is where we store configuration setting.

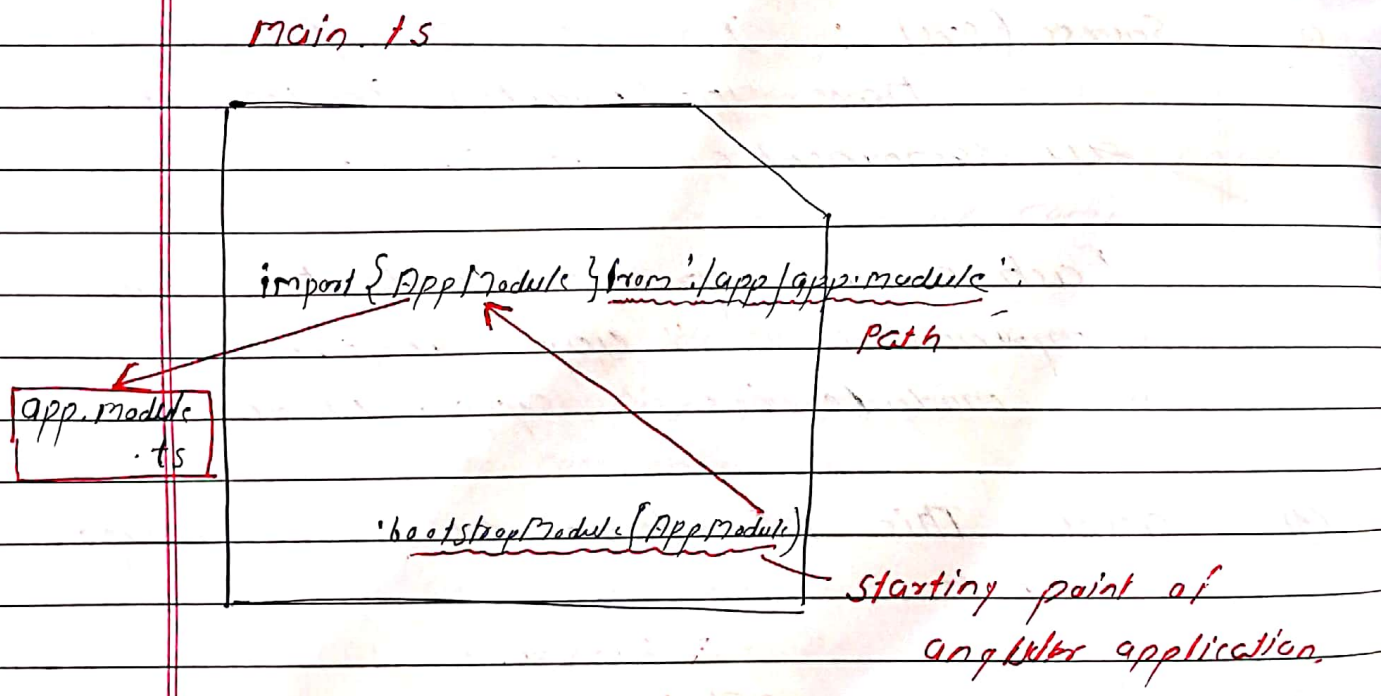
6) favicon.ico: In this file we can place icon. By default angular icon is present in it.

7) index.html: Simple HTML file containing our angular application.
   a) index.html page is responsible to display it on the browser.

(9) main.ts : • It is a typescript file. This is the starting point of our application (just like main method in Java)

(5) styles.css : We add global styles (more than one place) for our application.

(10) Webpack : Angular CLI uses a tool called webpack which is a build automation tool.



App component is registered by itself.

i) main.ts is an entry point it begins from AppModule, once it started it's booting it will check whether AppModule is imported or not

ii) If it is imported the path will be right beside it ('./app/app.module';)

iii) It will redirect to app.module.ts page

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

(decorator) ← @NgModule({

declarations: [AppComponent], → (Register component here)

imports: [BrowserModule, AppRoutingModule] → (Modules)

providers: []

(starts from here) ← bootstrap: [AppComponent]

app.component.css
app.component.html
app.component.spec.ts
app.component.ts

export class AppModule { }

app.module.ts is a default module which has @NgModule it is a decorator which has declarations, imports, providers and bootstrap.

Declaration has all the component. By default AppComponent is registered over here.

We can create multiple of components every component has to be registered here.

Imports: all the modules will be registered over here

AppRoutingModule it is used for Routing (without re-freshing the page the content has to be changed.

providers: Inside providers the service reference will be passed here.

bootstrap: (AppComponent): This is where the beginning of the app component.

AppComponent:-

This is the default component which is registered.

Components: Component is a major building block of a webpage of angular application. It will control the webpage.

- Component is a class where we can declare variables, function and we can also write business logic.
- Every component will have 4 file.

a) HTML file/page (Template) :
b) CSS file (It is used for styling purpose)
c) spec.ts (It is used for testing)
d) .ts (It is used for writing the Typescript code (Javascript)

It has class wherein we can write all the javascript code.

Note: Whenever the component will be activated first it will go to typescript file for example app.component.ts,

AppComponent

i) app.component.ts

```
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent
{
  title = 'HelloApp';
}
```

In app.component.ts file there is @Component decorator which has

① selector: 'app-root' (app-root is unique id (tag))

(b) templateUrl : In templateUrl our html link will be available.

(c) styleUrls : The style url (link) is present.

The class is present in app.component.ts where we can declare variable, function and we can write business logic.
All the javascript code we will write in side class.

i) (app.component.html)

```
<h1> Welcome </h1>
<h3> We are in app.component.html </h3>
```

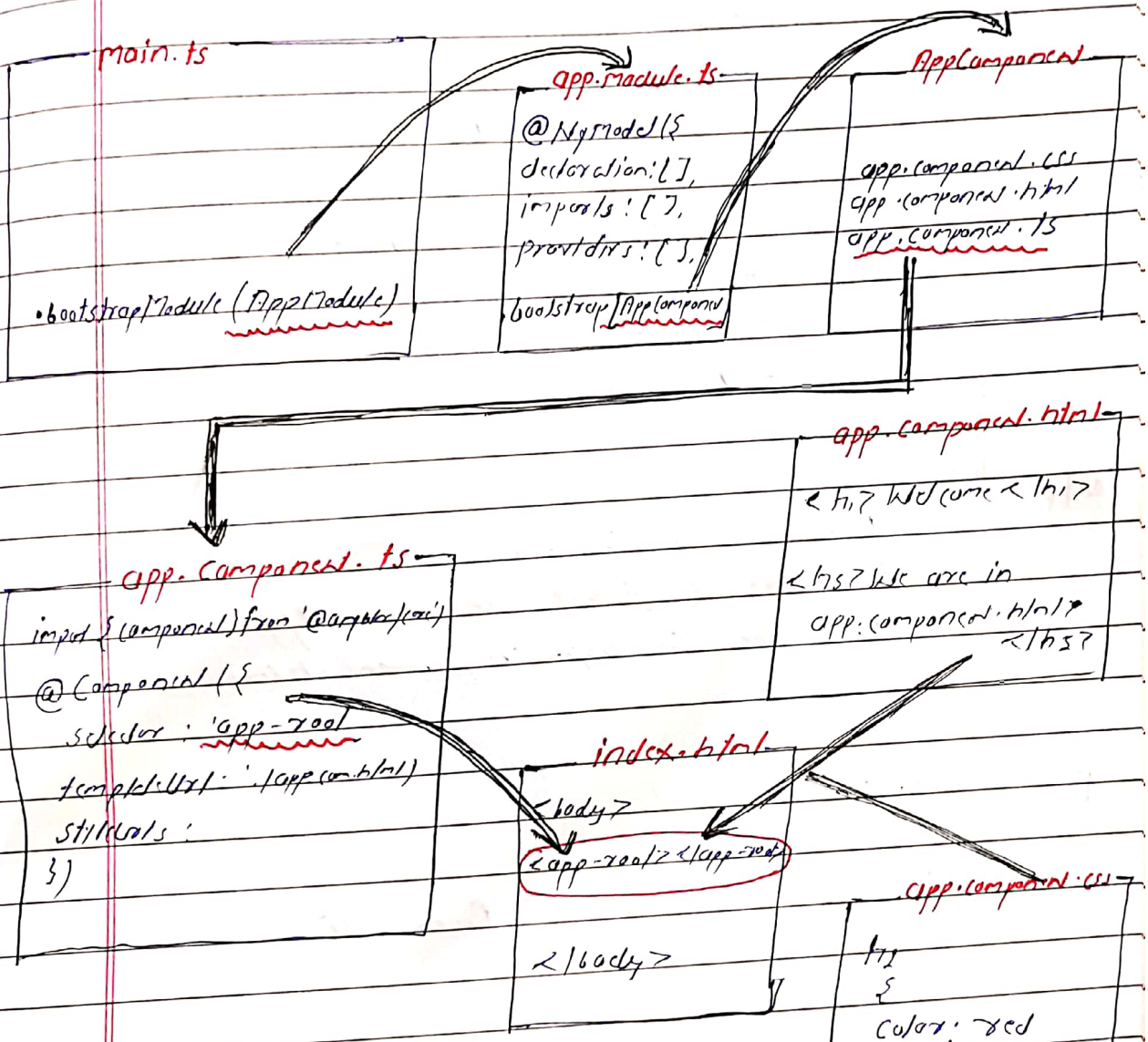
Index.html page

```
<html>
  <head>
  </head>
<body>
<app-root> </app-root>
</body>
</html>
```

app.component.css
app.componin.html
app.component.spec.ts
app.component.ts

Index.html page is responsible for displaying the entire code on the browser <app-root> </app-root> it is an unique tag (selector) which is responsible for activating the AppComponent

Execution - Flow



angular.json

[~~1 to 16 line~~ 8, 9, 17, 18 th line]